MEMORY HAVING INCREASED DATA-TRANSFER SPEED AND RELATED SYSTEMS AND METHODS

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[1] The invention relates generally to semiconductor circuits, and more particularly to a memory that allows faster data transfers than similar conventional memories, to systems that incorporate the memory, and to related data-transfer methods. In one embodiment, the memory generates an internal column address and receives an external column address during a page-mode data-transfer (read/write) cycle. By coupling the internal address to the column decoder instead of the external address, the memory can achieve a faster data-transfer speed. And by monitoring the external address, the memory allows the system to access a page of any length.

DESCRIPTION OF THE BACKGROUND ART

[2] The operating speeds of processor peripherals such as memory circuits often prevent engineers from designing faster digital-electronic systems. The speeds of microprocessors, which are at the hearts of today's digital systems, have increased dramatically within the last few years. But the speeds of today's memory circuits and other processor peripherals have lagged behind. Therefore, these slower peripherals typically limit the overall speed of a digital system because the system microprocessor must effectively "slow down" during transfers of data to and from these peripherals. That is, these slower peripherals are the "weak link in the chain".

[3] **FIG. 1** is a timing diagram of a conventional fast-page-mode read cycle, which increases a memory's (memory not shown in **FIG.** 1) data-transfer speed by allowing a system (not shown in **FIG. 1**) to read multiple columns within a row, *i.e.*, "page," without reasserting the row address between reads. At time t_0, the system asserts an active-low Row Address Strobe ($\overline{RAS}$) to latch the row address that the system has driven onto an external address bus. At time t_1, the system drives the first column address onto the external address bus, and, at time t_2, the system

asserts an active-low Column Address Strobe ($\overline{CAS}$) to latch the first column address. At time t_3, the data stored in the addressed memory location (not shown in **FIG. 1**) appears on the data bus. The system repeats these steps for subsequent column addresses until it reads data from all the desired memory locations within the addressed row. To exit the fast-page mode, the system transitions ($\overline{RAS}$) to an inactive high level (transition not shown in **FIG. 1**).

[4] Unfortunately, latching an external column address for each data transfer limits the minimum fast-page-mode cycle time t_{pc}, which is the minimum period of $\overline{CAS}$ that the memory manufacturer specifies for proper operation of the memory in fast-page mode. Each column address requires a minimum time t_s — sometimes called the setup or precharge time — to propagate from the external address bus, through the memory's front-end circuitry (not shown in **FIG.1**), to the column-address decoder (not shown in **FIG. 1**), where the system latches the column address by asserting $\overline{CAS}$. Therefore, to prevent data-transfer errors, t_{pc} must be long enough to account for t_s. If, however, t_{pc} is not long enough to account for t_s, then the column-address decoder may latch an erroneous column address when the system asserts $\overline{CAS}$; consequently, the system may end up reading data from the wrong memory location.

[5] **FIG. 2** is a timing diagram of a conventional nibble-mode read cycle, which allows faster data transfers than the fast-page-mode read cycle (**FIG. 1**) does because the memory (not shown in **FIG.2**) generates the column addresses internally. As in the fast-page mode, the system (not shown in **FIG. 2**) asserts $\overline{RAS}$ at time t_0 to latch the row address. At time t_1, the system drives a first column base address onto the external address bus, and, at time t_2, the system asserts $\overline{CAS}$ to latch this base address. An adder (not shown **in FIG. 2**) inside the memory sums a nibble count (here 00) with the base address to generate an initial column address, and, at time t_3, the data stored at the generated column address appears on the data bus. Next, a nibble counter (not shown in **FIG. 2**) inside of the memory increments the nibble count, and the adder sums the incremented nibble count (here 01) with the base address to generate a first subsequent column address (not shown in **FIG. 2**).

At time t_4, the system asserts $\overline{CAS}$ to latch this subsequent column address into the address decoder (not shown in **FIG. 2**), and at time t_5, the data stored in the addressed memory location appears on the data bus. The system continues asserting $\overline{CAS}$ until it has accessed a predetermined number — sometimes called a "nibble" — of columns (here four columns) within the row. To address a second nibble of columns within the same row, the system drives a second column base address (not shown in **FIG. 2**) onto the address bus and clocks $\overline{CAS}$ to repeat the cycle. Because the memory generates the subsequent column addresses internally, the minimum nibble-mode cycle time t_{nc} (for all $\overline{CAS}$ cycles except the initial cycle) need not account for the external-address setup time t_s (**FIG. 1**); consequently, t_{nc} is typically shorter than the minimum fast-page-mode cycle time t_{pc}. In one example, t_{pc} is 60 nanoseconds (ns), and t_{nc} for the same memory is 40 ns. Thus, the system can read data from the memory faster during the nibble mode than it can during the fast-page mode.

[6] Unfortunately, because the number of columns in a nibble is typically fixed, reading data from more than one nibble of columns within a row often slows the data transfer by requiring the system to drive an additional column index address onto the external address bus for each additional nibble to be read. The nibble counter within the memory (neither shown in **FIG. 2**) typically has a fixed length (here two bits), therefore, if the system wants to read more than one nibble's worth (four columns here) of data from a row, it must execute multiple nibble-mode cycles. As stated above, the system must drive a column index address onto the external address bus at the beginning of each nibble-mode cycle. Because the column index address is on the external address bus, a setup time approximately equal to t_s is required for the index address to propagate from the external address bus, through the memory's front-end circuitry, to the nibble adder (not shown in **FIG.1**), and for the generated column address to propagate from the adder to the column-address decoder. Thus, the minimum nibble-mode cycle time for the first cycle is significantly longer than t_{nc}, and may be as long or longer than the minimum page-mode cycle time t_{pc}.

[7] Furthermore, because a memory often does not allow the system to read a partial nibble, the number of columns the system reads often must be divisible by the number of columns in a nibble. For example, if there are four columns in a nibble, the number of columns the system can access during a nibble-mode cycle must be a multiple of four, *i.e.,* 4, 8, 12, 16, . . .

[8] Although fast-page- and nibble-mode read cycles are discussed above in conjunction with **FIGS. 1** and **2**, fast-page- and nibble-mode write cycles have similar problems.

SUMMARY OF THE INVENTION

[9] In one aspect of the invention, a memory includes an address bus, address counter, address decoder, comparator, and control circuit. During a data read or write cycle, the address bus receives an external address and the address counter generates an internal address. The address decoder decodes the internal address, and the comparator compares the external address to a value. Based on the relationship between the external address and the value, the comparator enables or disables the data transfer.

[10] For example, such a memory can terminate a page-mode cycle by determining when the current external column address is no longer equal to the current internal column address. This allows the system to terminate the cycle after any number of data transfers by setting the external column address to a value that does not equal the internal column address. Or, the memory can terminate the cycle by comparing the external address to a predetermined stop address.

[11] In a related aspect of the invention, the control circuit enables or disables the data transfer based on a relationship between the internal address and a programmable value.

[12] For example, the memory can terminate a page-mode cycle by comparing the internal column address to a programmable stop value, or by decrementing a programmable count value until the count reaches a predetermined stop value such as zero.

[13] Consequently, such a memory combines the higher data-transfer speed of a conventional nibble mode with the greater page-length flexibility of a conventional fast-page mode.

BRIEF DESCRIPTION OF THE DRAWINGS

[14] **FIG. 1** is a timing diagram of a conventional fast-page-mode read cycle.

[15] **FIG. 2** is a timing diagram of a conventional nibble-mode read cycle.

[16] **FIG. 3** is a timing diagram of a page-mode read cycle according to an embodiment of the invention.

[17] **FIG. 4** is a schematic block diagram of a page-mode circuit that can execute the page-mode read cycle of **FIG. 3** according to an embodiment of the invention.

[18] **FIG. 5** is a schematic block diagram of a memory circuit that includes the page-mode circuit of **FIG. 4** according to an embodiment of the invention.

[19] **FIG. 6** is a schematic block diagram of a digital computer system that includes the memory circuit of **FIG. 5** according to an embodiment of the invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[20] The following discussion is presented to enable one skilled in the art to make and use the invention. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the generic principles herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention as defined by the appended claims. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

[21] **FIG. 3** is a timing diagram of a page-mode read cycle according to an embodiment of the invention. During this cycle, a memory (**FIG. 5**) generates an internal column address to increase the data-transfer speed as it would if executing the nibble-mode read cycle of **FIG. 2**. But in addition, the memory receives an external column address, and, as discussed below in conjunction with **FIGS. 3** and **4**,

monitors the external column address to determine when to terminate the cycle. Therefore, unlike the nibble-mode read cycle, the improved page-mode read cycle allows the memory to latch only one column address regardless of the length of the page, *i.e.,* the number of columns addressed within the addressed row. Consequently, the improved page-mode cycle time t_{ipc} is often shorter than the cycle time t_{pc} (**FIG. 1**), and, in one embodiment, is the same or approximately the same as the nibble-mode cycle time t_{nc} (**FIG. 2**).

[22] **FIG. 4** is a schematic block diagram of a page-mode circuit ***12*** that allows a memory (**FIG. 5**) to execute the page-mode cycle of **FIG. 3** according to an embodiment of the invention. The circuit ***12*** includes a column-address-anticipation counter ***14*** that receives an initial column address or base address from the external address bus and increments, decrements, or otherwise varies this initial address to generate an internal column address in response to the rising edge of $\overline{CAS}$. An optional page-length register/counter ***16*** stores and/or generates a page-stop value, and a comparator ***18*** of a mux/comparator circuit ***20*** generates a comparison signal having a value that is based on a comparison of two of the following values: the external column address, internal column address, and the page-stop value. A multiplexer ***22*** of the mux/comparator circuit ***20*** couples the internal column address to the column decoder (**FIG. 5**) during the page-mode cycle, and may couple the external column address to the column decoder in other data-transfer modes. A control circuit ***24*** controls the counter ***14***, register/counter ***16***, and the mux/comparator circuit ***20*** in response to the comparison signal from the comparator ***18*** and from signals received from the system (**FIG. 6**). For example, the control circuit ***24*** can configure the column address counter ***14***, the register/counter***16***, or both to increment or decrement based on the value of an Increment/Decrement signal from the system. Also, the control circuit ***24*** can generate a data-transfer enable/disable signal to enable the data buffers and/or the counter ***14*** (**FIG. 5**) during a page-mode cycle and to disable the data buffers and/or the counter ***14*** when the system terminates the cycle.

[23] **FIG. 5** is a block diagram of a memory circuit ***26***, which includes the page-mode circuit ***12*** of **FIG. 4** according to an embodiment of the invention. In the

disclosed embodiment, the memory ***26*** is a Dynamic Random Access Memory (DRAM), although the memory ***26*** may be another type of memory such as a Static Random Access Memory (SRAM). Furthermore, other than the circuit ***12***, the other blocks of the memory ***26*** are conventional, and, therefore, are not discussed in detail.

[24] In addition to the page-mode circuit ***12***, the memory circuit ***26*** includes a memory array ***30*** for storing data, a data-in buffer ***32*** for receiving write data that the system (**FIG. 6**) drives onto the data bus during a write cycle, and a data-out buffer ***34*** for driving read data from the array ***30*** onto the data bus during a read cycle such as the page-mode read cycle of **FIG. 3**. A row decoder ***36*** activates the addressed row of the memory array ***30*** to be written to or read from, and a column decoder ***38*** couples the addressed column or columns to the buffer ***32*** (write cycle) or the buffer ***34*** (read cycle) via sense amplifiers ***40***. A row-address buffer ***42*** stores the row address that the system drives onto the address bus and provides it to the row decoder ***36***. Likewise, a column-address buffer ***44*** stores the column address that the system drives onto the address bus and provides it to the column decoder ***38*** via the multiplexer ***22*** (**FIG. 4**) of the mux/comparator circuit ***20***. A logic gate such as a NAND gate ***46*** enables/disables the data-in and data-out buffers ***32*** and ***34*** in response to the system read/write and $\overline{CAS}$ signals. Clock generators ***48*** and ***50*** generate respective clock signals for the circuit blocks of the memory ***26***, and a refresh circuit ***52*** periodically refreshes the data stored in the memory array ***30***.

[25] Referring to **FIGS. 3 – 5**, the operation of the memory circuit ***26*** during the page-mode read cycle of **FIG. 3** is discussed according to an embodiment of the invention.

[26] At time t_0, the system (**FIG. 6**) asserts $\overline{RAS}$ to latch the row address, ROW 0, in the row-address buffer ***42***. Also, by asserting the appropriate value for Inc/Dec, the system instructs the control circuit ***24*** to configure the column-address anticipation counter ***14*** to operate in increment mode.

[27] At time t_1, the system drives the initial column address COL 0 onto the address bus, and in response, the column-address counter ***14*** stores COL 0.

[28] At time t_2, the system asserts $\overline{CAS}$ to latch the address COL 0 in the counter ***14***. COL 0 propagates to the column decoder ***38***, which causes the sense amplifiers ***40*** to load the data, DATA 0, stored at ROW 0, COL 0 of the memory array **30** into the data-out buffer ***34***.

[29] At time t_3, the clock generator ***48*** clocks DATA 0 from the data-out buffer ***34*** onto the data bus.

[30] Also at time t_4, the system transitions $\overline{CAS}$ to an inactive level, and, in response to this transition, the counter ***14*** increments the internal column address to COL 1 at time t_5. COL 1 is the column address that the counter ***14*** "anticipates" to be the next external column address that the system will drive onto the address bus.

[31] At time t_6, the system drives the external column address COL 1 onto the address bus. The comparator ***18***, under the control of the control circuit ***24***, compares the external column address on the address bus to the internal column address in the counter ***14***. Because these addresses are equal, the control circuit ***24*** determines that the page-mode read cycle is still active. Also, the internal column address COL 1 propagates from the counter ***14***, through the multiplexer ***22***, to the column decoder ***38***. Because COL 1 is internally generated, the setup time — the time required for COL 1 to propagate from the counter ***14*** to the column decoder ***38*** — is shorter than it would be for the external generated COL 1. Furthermore, the counter ***14*** generates internal COL 1 before the system drives external COL 1 onto the address bus. Therefore, the combination of these two factors allows the internal COL 1 to arrives at the column decoder ***38*** sooner than the external COL 1. Consequently, for the same memory circuit, the page-mode cycle time t_{ipc} is typically much shorter than t_{pc} of the convention fast-page mode of **FIG. 1**, on the order of t_{nc} of **FIG. 2**.

[32] At time t_7, the system asserts $\overline{CAS}$, and the column decoder ***38*** causes the sense amplifiers ***40*** to load the data, DATA 1, stored at ROW 0, COL 1 of the memory array ***30*** into the data-out buffer ***34***.

[33] At time t_8, the clock generator ***48*** clocks DATA 1 from the data-out buffer ***34*** onto the data bus.

[34] At time t_9, the system transitions $\overline{CAS}$ to an inactive level, and, in response to this transition, the counter ***14*** increments the internal column address to COL 2 at time t_{10}. COL 2 is the column address that the counter ***14*** anticipates to be the next external column address that the system will drive onto the address bus.

[35] At time t_{11}, the system drives the column address COL 2 onto the address bus. The comparator ***18***, under the control of the control circuit ***24***, compares the external column address on the address bus to the internal column address in the counter ***14***. Because these addresses are equal, the control circuit ***24*** determines that the page-mode read cycle is still active. Also, the internal column address COL 2 propagates from the counter ***14***, through the multiplexer ***22***, to the column decoder ***38***.

[36] At time t_{12}, the system asserts $\overline{CAS}$, and the column decoder ***38*** causes the sense amplifiers ***40*** to load the data, DATA 2, stored at ROW 0, COL 2 of the memory array **30** into the data-out buffer ***34***.

[37] At time t_{13}, the clock generator ***48*** clocks DATA 2 from the data-out buffer ***34*** onto the data bus.

[38] At time t_{14}, the system transitions $\overline{CAS}$ to an inactive level, and, in response to this transition, the counter ***14*** increments the column address to COL 3 at time t_{15}. COL 3 is the address that the counter ***14*** anticipates to be the next external column address that the system will drive onto the address bus.

[39] At time t_{16}, the system drives the external column address COL 2, or any value other than COL 3, onto the address bus to signal the end of the page-mode read cycle. The comparator ***18*** compares the valve on the address bus to the internal column address in the counter ***14***. Because the valve and address are unequal, the control circuit ***24*** determines that the system has terminated the page-mode read cycle. Therefore, the control circuit ***24*** disables the data-out buffer ***34***, the counter ***14***, or both, and the memory ***26*** awaits the next system command.

[40] Consequently, such a page-mode read cycle is faster than the fast-page-mode cycle of **FIG. 1**, and is at least as fast, but less restrictive, than the

nibble-mode cycle of **FIG. 2**. Because the multiplexer ***22*** couples the internal column address, not the external column address, to the column decoder ***38***, t_{ipc} is shorter than t_{pc}. Therefore, the memory ***26*** can execute the page-mode read cycle of **FIG. 1** faster than it can execute the fast-page-mode read cycle of **FIG. 2**. For the same reason, t_{ipc} is often approximately the same as t_{nc} of **FIG. 2**. Furthermore, if the system reads more than four columns of data from the row, the memory ***26*** can execute the page-mode read cycle of **FIG. 3** faster than it can execute the nibble-mode read cycle of **FIG. 2** because the memory need only load one external column address or index address to execute the cycle of **FIG. 3**.

[41] Although an embodiment of the page-mode read cycle of **FIG. 3** discussed, one can design the memory ***26*** to implement a page-mode write cycle in a similar fashion. Furthermore, although the cycle is discussed in terms of reading multiple columns within a row, one can design the memory ***26*** to allow the reading of multiple rows within a column. Moreover, instead of generating the internal column address, the column-address anticipation counter ***14*** may generate an index that an adder (not shown) adds to a base address (COL 0 in the above example) to generate the column address.

[42] Still referring to **FIGS. 3 – 5**, other embodiments of the page-mode read cycle of **FIG. 3** are envisioned.

[43] In one embodiment, the page-mode read cycle is as described above except that the comparator ***18*** compares the external column address to the contents of the length register/counter ***16***. The system loads the ending column, here COL 3, into the length register/counter ***16*** before commencing the read cycle. Next, the system loads the initial column address COL 0 into the column address anticipation counter ***14***, and the cycle proceeds as discussed above, except that the comparator ***18*** compares the external column address to the contents of the register/counter ***16*** instead of to the internal column address. As long as the external column address does not equal COL 3, the control circuit ***24*** determines that the page-mode cycle is still active. But when the external column address equals COL 3, the control circuit ***24*** terminates the page-mode read cycle. Therefore, the control circuit ***24*** disables, the data-out buffer ***34*** and/or the counter **14**, and the memory ***26***

awaits the next system command. Although in this embodiment the memory ***26*** would not output data stored in the ending column, here COL 3, one can modify the memory ***26*** so that it would output data from the ending column.

[44] In a similar embodiment, the comparator ***18*** compares the internal column address from the column-address anticipation counter ***14*** to the cycle-ending address stored in the length register/counter ***16***.

[45] In yet another embodiment, the length register/counter ***16*** functions as a counter, and the control circuit ***24*** terminates the page-mode cycle when the count reaches a predetermined value. The system loads the register/counter ***16*** with the number, here three, of columns to be read (COL 0 – COL 2) on the first assertion of $\overline{CAS}$. The register/counter ***16*** then decrements the number by one in response to each subsequent assertion of $\overline{CAS}$. When the contents of the register/counter ***16*** equals zero (or some other predetermined end value), the control circuit ***24*** terminates the page-mode read cycle. Although described as decrementing the number of columns, the register/counter ***16*** can start from zero or another predetermined starting value and increment the number. When the number equals an ending number that the system has programmed into the control circuit ***24*** or into another block of the memory ***26***, the circuit ***24*** terminates the page-mode read cycle.

[46] **FIG. 6** is a block diagram of an electronic system ***60***, such as a computer system, that incorporates the memory circuit ***26*** of **FIG. 5** according to an embodiment of the invention. The system ***60*** includes computer circuitry ***62*** for performing computer functions, such as executing software to perform desired calculations and tasks. The circuitry ***62*** typically includes a processor ***64*** and the memory circuit ***26***, which is coupled to the processor ***64***. One or more input devices ***66***, such as a keyboard or a mouse, are coupled to the computer circuitry ***62*** and allow an operator (not shown) to manually input data thereto. One or more output devices ***68*** are coupled to the computer circuitry ***62*** to provide to the operator data generated by the computer circuitry ***62***. Examples of such output devices ***68*** include a printer and a video display unit. One or more data-storage devices ***70*** are coupled to the computer circuitry ***62*** to store data on or retrieve data from external storage media (not shown). Examples of the storage devices ***70*** and the

corresponding storage media include drives that accept hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs). Typically, the computer circuitry ***62*** includes the address, data, and control buses that are coupled to the memory circuit ***26***.